AMENDMENTS TO THE CLAIMS

Please amend claims 1, 10, 13, 23, and 33; and add new claim 35, as shown below. All pending claims are reproduced below, including those that remain unchanged.

1. (Currently Amended) A computer-enabled system embodied in a storage medium to provide a software debugging environment, comprising:

an executing software program containing at least one data structure;

at least one abstract view capable of displaying and/or editing at least one abstract content of the at least one data structure, wherein the abstract content of the at least one data structure constitutes attributes of interest during the execution of the executing software program rather than underlying physical data structures used to represent the abstract content; and

at least one filter capable of extracting and formatting the contents of interest from the underlying physical data structures and defining a displaying and/or editing property of the at least one abstract view, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited, wherein the at least one filter is used to extract data from and manipulate one or more contents of buffer used to transmit and receive streaming data.

2. (Original) The system according to claim 1, wherein:

the system is at least partially implemented using Java language.

3. (Original) The system according to claim 1, further comprising:

at least one editor associated with the at least one abstract view capable of at least one of:

allowing the at least one abstract content to be modified through the at least one abstract view; and

validating an input value to the at least one abstract content against an allowed value for the at least one abstract content.

4. (Previously Presented) The system according to claim 1, further comprising:

at least one another abstract view is capable of presenting the at least one another abstract content of the at least one data structure without showing a physical implementation of the at least one data structure.

5. (Original) The system according to claim 1, wherein:

each of the at least one abstract view can be individually selected for display.

6. (Original) The system according to claim 1, wherein:

Two or more of the at least one abstract view are capable of displaying and/or editing the same one of the at least one abstract content without being deadlocked.

7. (Previously Presented) The system according to claim 1, wherein:

the at least one filter can be defined via configuration information stored in a file, which can be a file in a markup language.

8. (Original) The system according to claim 1, further comprising:

a component capable of interactively performing at least one of:

selecting a subset of the at least one of abstract view for display; and

defining the displaying and/or editing property of the at least one filter.

9. (Original) The system according to claim 8, wherein:

the component can be realized via an interface to an Integrated Development Environment (IDE).

10. (Currently Amended) The system according to claim 1, further comprising:

at least one component capable of supporting the debugging of a ~~JSP~~ server page and a machine generated servlet that implements the ~~JSP~~ server page.

11. (Previously Presented) The system according to claim 10, wherein:

the at least one component can perform at least one of:

extracting and displaying a code and/or a content of interest, and mapping them to a format used in a source code in a server page, for use with executing a servlet;

following an execution path through at least one level of redirection using at least one tag;

extracting and manipulating a streaming data from a content of a buffer used to transmit and receive the streaming data; and

setting at least one break point in a server page and stepping through the execution of the page based on the displaying property.

12. (Previously Presented) The system according to claim 11, wherein:

the streaming data can be extracted by inserting a wrapper or "writer" class around the servlet.

13. (Currently Amended) A method to provide a software debugging environment, comprising:

displaying and/or editing at least one abstract content of at least one data structure in an executing software program via at least one abstract view, wherein the abstract content of the at least one data structure constitutes attributes of interest during the execution of the executing software program rather than underlying physical data structures used to represent the abstract content; and

extracting and formatting, via at least one filter, the contents of interest from the underlying physical data structures and defining a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited, wherein the at least one filter is used to extract data from and manipulate one or more contents of buffer used to transmit and receive streaming data.

14. (Original) The method according to claim 13, further comprising:

allowing the at least one abstract content to be modified through the at least one abstract view; and

validating an input value to the at least one content against an allowed value for the at least one content.

15. (Original) The method according to claim 13, further comprising:

presenting the at least one abstract content of the at least one data structure without showing a physical implementation of the at least one data structure.

16. (Original) The method according to claim 13, further comprising:

selecting each of the at least one abstract view individually for display.

17. (Original) The method according to claim 13, further comprising:

displaying and/or editing the same one of the at least one abstract content via two or more of the at least one abstract view without being deadlocked.

18. (Previously Presented) The method according to claim 13, further comprising:

defining the at least one filter via configuration information stored in a file, which can be a file in a markup language.

19. (Original) The method according to claim 13, further comprising:

interactively performing at least one of:

selecting a subset of the at least one of abstract view for display; and

defining the displaying and/or editing property of the at least one filter.

20. (Previously Presented) The method according to claim 13, further comprising:

supporting the debugging of a server page and a machine generated servlet that implements the server page.

21. (Previously Presented) The method according to claim 20, further comprising:

extracting and displaying a code and/or a content of interest, and mapping them to a format used in a source code in a server page, for use with executing a servlet;

following an execution path through at least one level of redirection using at least one tag;

extracting and manipulating a streaming data from a content of a buffer used to transmit and receive the streaming data; and

setting at least one break point in a server page and stepping through the execution of the page based on the displaying property.

22. (Original) The method according to claim 21, wherein:

the streaming data can be extracted by inserting a wrapper or "writer" class around the JSP servlet.

23. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

display and/or edit at least one abstract content of at least one data structure in an executing software program via at least one abstract view, wherein the abstract content of the at least one data structure constitutes attributes of interest during the execution of the executing software program rather than underlying physical data structures used to represent the abstract content; and

extract and format, via at least one filter, the contents of interest from the underlying physical data structures and defining a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited, wherein the at least one filter is used to extract data from and manipulate one or more contents of buffer used to transmit and receive streaming data.

24. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

allow the at least one abstract content to be modified through the at least one abstract view; and

validate an input value to the at least one abstract content against an allowed value for the at least one content.

25. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

> present the at least one abstract content of the at least one data structure without showing an physical implementation of the at least one data structure.

26. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

> select each of the at least one abstract view individually for display.

27. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

> display and/or edit the same one of the at least one abstract content via two or more of the at least one abstract view without being deadlocked.

28. (Previously Presented) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

> define the at least one filter via configuration information stored in a file, which can be a file in a markup language.

29. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

> interactively perform at least one of:
>
> selecting a subset of the at least one of abstract view for display; and
>
> defining the displaying and/or editing property of the at least one filter.

30. (Previously Presented) The machine readable medium of claim 23 further comprising instructions that when executed cause the system to:

> support the debugging of a server page and a machine generated servlet that implements the server page.

31. (Previously Presented)   The machine readable medium of claim 30, further comprising instructions that when executed cause the system to:

extract and display a code and/or a content of interest, and mapping them to a format used in a source code in a server page, for use with executing a servlet;

follow an execution path through at least one level of redirection using at least one tag;

extract and manipulate a streaming data from a content of a buffer used to transmit and receive the streaming data; and

set at least one break point in a server page and step through the execution of the page based on the displaying property.

32. (Previously Presented) The machine readable medium of claim 31, wherein:

the streaming data can be extracted by inserting a wrapper or "writer" class around the servlet.

33. (Currently Amended) A computer-enabled system embodied in a storage medium to provide a software debugging environment, comprising:

means for displaying and/or editing at least one abstract content of at least one data structure in an executing software program via at least one abstract view, wherein the abstract content of the at least one data structure constitutes attributes of interest during the execution of the executing software program rather than underlying physical data structures used to represent the abstract content; and

means for extracting and formatting, via at least one filter, the contents of interest from the underlying physical data structures and defining a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited, wherein the at least one filter is used to extract data from and manipulate one or more contents of buffer used to transmit and receive streaming data.

34. (Canceled).

35. (New) A computer-enabled system embodied in a storage medium to provide a JAVA Server Page (JSP) software debugging environment, comprising:

a JSP source code to be debugged associated with a tag library;

at least one abstract view capable of displaying and/or editing at least one abstract content of at least one data structure, wherein the abstract content of the at least one data structure constitutes attributes of interest associated with the JSP source code to be debugged based on the underlying physical data structures during the execution of the one or more machine generated JAVA code; and

at least one filter capable of extracting and formatting the contents of interest from the underlying physical data structures and defining a displaying and/or editing property of the at least one abstract view, wherein the at least one filter allows a user to select which method in the tag library should be hit when stepping into the tag library from the JSP source code.